



# Techniques of Water-Resources Investigations of the United States Geological Survey

## Chapter A5

### A MODULAR FINITE-ELEMENT MODEL (MODFE) FOR AREAL AND AXISYMMETRIC GROUND-WATER-FLOW PROBLEMS, PART 3: DESIGN PHILOSOPHY AND PROGRAMMING DETAILS

By Lynn J. Torak

Book 6  
MODELING TECHNIQUES

Table 6.—Subroutines that set dimensions, allocate storage, and initialize variables

Subroutine	Description of computations	Versions of MODFE that require subroutine								
		LMFE1 (LMFE2)	LMFE3 (LMFE4)	NLMFE1 (NLMFE2)	NLMFE3 (NLMFE4)	NLMFE5 (NLMFE6)	NLMFE7 (NLMFE8)	NSSF1 (NSSF2)	NSSF3 (NSSF4)	NSSF5 (NSSF6)
CBINIT	Allocate storage for transient leakage terms and initialize storage locations to zero.		x (x)		x (x)					
GNINIT	Allocate storage for $\alpha$ coefficients, node numbers, and controlling heads and altitudes for nonlinear head-dependent (Cauchy-type) boundaries and nonlinear point sinks. Initialize storage locations to zero.						x (x)			x (x)
INITB	Set dimensions for storage vectors, allocate storage for condensed matrix, reduced matrix, element areas and incidences, nodal coordinates, known fluxes, hydraulic head, boundary or external head, $\alpha$ coefficient, specified flux, boundary-node numbers, source-bed head, boundary-condition zone numbers, time steps, and node-indicator vector. Initialize time and flow-balance terms and storage locations for above terms to zero.	x	x	x	x	x	x	x	x	x
INITCG	Same as INITB, and allocate storage for vectors X, P, and R for MICCG method.	(x)	(x)	(x)	(x)	(x)	(x)	(x)	(x)	(x)
SWINIT	Allocate storage for aquifer thickness and altitude of top of aquifer, and initialize storage locations to zero.							x (x)	x (x)	x (x)
VNINIT	Allocate storage for hydraulic conductance term and altitudes for nonlinear steady vertical leakage. Initialize storage locations to zero.					x (x)			x (x)	
WTINIT	Allocate storage for head-change vector, aquifer thickness, altitude of top of aquifer, and specific yield for water-table simulations. Initialize storage locations to zero.			x (x)	x (x)	x (x)	x (x)	x (x)	x (x)	x (x)

## Areally Distributed Sources and Sinks

The term for areally distributed sources and sinks expressed as  $(1/3)W^e\Delta^e$  in equation (50) in Cooley (1992) for element  $e$  is computed in subroutine FMCO. This term represents the element contribution to the areally distributed flux at a node, which is part of the B vector of equations (1) and (2). Values for the volumetric flow rate per unit area [length/time], or the unit rate, are input by hydraulic-property zone as the program variable QD. The element contribution to areally distributed recharge or discharge expressed in equation (50) in Cooley (1992) is formed by computations similar to those performed for the storage-coefficient term: the unit rate is multiplied by the square of the scaling factor SCALE, divided by 6, and multiplied by twice the element area (program variable AREA see section "Storage Coefficient"). The resulting term is represented by program variable TEQ, and is added to the B vector of equations (1) and (2) for each node in the element. The B vector is

represented by program vector Q. These computations are performed for all elements in the hydraulic-property zone and for all zones, thus satisfying the summation of areally distributed terms over all elements, as indicated by equation (50) in Cooley (1992).

The TEQ terms are summed in subroutine FMCO according to sign—positive for recharge—to obtain the terms used in the water-balance summary for areally distributed sources and sinks. It is convenient to perform the summation of TEQ terms at this location in MODFE as these terms are no longer needed after they are added into storage locations in the Q vector. The terms used in the water-balance summary for areally distributed sources and sinks are represented by DQI for recharge and DQO for discharge.

## Point Sources and Sinks

Computations for incorporating point sources and sinks into matrix equations (1) and (2) are performed in subroutine DATIN. The volumetric rate [length<sup>3</sup>/

Table 7.—Subroutines that input hydrologic information and form coefficients to matrix equations

Subroutine	Description of input or coefficients (numbers in parentheses correspond to equations in Cooley, 1992)	Versions of MODFE that require subroutine								
		LMFE1 (LMFE2)	LMFE3 (LMFE4)	NLMFE1 (NLMFE2)	NLMFE3 (NLMFE4)	NLMFE5 (NLMFE6)	NLMFE7 (NLMFE8)	NSSF1 (NSSF2)	NSSF3 (NSSF4)	NSSF5 (NSSF6)
DATIN	Input indicators for axi-symmetric radial flow, scale factor and title, indicators to suppress or print selected input, nodal coordinates, hydraulic head, source bed head, and linear boundary conditions. Add volumetric flow rate from point sources or sinks to vector of known terms, (50). Evaluate vectors that store node indicators, mean head, and mean head change.	x (x)	x (x)	x (x)	x (x)	x (x)	x (x)	x (x)	x (x)	x (x)
CBFMCO	Input zone values for vertical hydraulic conductivity and specific storage for transient leakage. Form transient leakage coefficients according to (167) and assign values to regression parameters.		x (x)		x (x)					
FMCO	Input hydraulic properties by zone for aquifer and confining bed and form coefficients for transmissivity, (39), (40), or (43), or hydraulic conductivity, (72); storage coefficient, (36); vertical hydraulic conductance and linear boundary terms, (37); and known fluxes, (50). Input element incidences.	x (x)	x (x)	x (x)	x (x)	x (x)	x (x)	x (x)	x (x)	x (x)
GNFMCO	Input indicators to suppress or print values for nonlinear head-dependent (Cauchy-type) boundaries and nonlinear point sinks, input values for nonlinear head-dependent (Cauchy-type) boundaries, and nonlinear point sinks. Form coefficients according to (155).						x (x)			x (x)
SWFMCO	Input indicators to suppress or print aquifer thickness or altitude of aquifer top, input aquifer thickness and altitude of aquifer top. Form transmissivity coefficients according to (244).							x (x)	x (x)	x (x)
VNFMCO	Input indicators to suppress or print values for nonlinear steady vertical leakage. Form coefficients according to (119) and (132).					x (x)			x (x)	
WTFMCO	Input aquifer thickness, altitude of top of aquifer, and specific yield for water-table simulations. Form coefficients according to (36) and (47).			x (x)	x (x)	x (x)	x (x)			

time] for point stresses are input by node as the program variable QWEL, and are added to the appropriate locations in the program vector Q that stores the B vector of the matrix equations.

The nodal values for point sources and sinks are not stored in MODFE after their incorporation into program vector Q; hence, computations for the water-balance summary pertaining to point sources and sinks are performed in subroutine DATIN. Values of QWEL are summed according to sign, positive for a point source, and stored as program variable WQI for point sources and WQO for point sinks.

### Specified-Head Boundaries

Inputs and computations involving specified-head boundaries occur in subroutines DATIN, EXTRAP, and XTRPWT. Values for specified-head boundaries

are input by node as the program variable HB in subroutine DATIN. The head difference,  $\delta_i$ , and average head,  $\bar{h}_i$ , at each specified-head node  $i$  are computed in subroutine DATIN according to equations (57) and (63) in Cooley (1992). The average head replaces the initial head at the node, stored in program vector H, and the average head difference is stored in program vector DHB. These values are used to form components of finite-element matrix equation (254) in Cooley (1992) for nodes that connect to the specified-head boundary. Details of the formation of finite-element matrix equations are given in Cooley (1992) and in the section "Reordering Finite-Element Equations for Solution."

The average head,  $\bar{h}_i$ , is replaced by the value of the specified head, and the head change,  $\delta_i$ , is set to zero following the first time step that uses the specified-head values. These computations occur in subroutine

Table 8.—Subroutines that input stress-period and time-step information and adjust boundary conditions for time-step variation

Subroutine	Description of input or computation	Versions of MODFE that require subroutine								
		LMFE1 (LMFE2)	LMFE3 (LMFE4)	NLMFE1 (NLMFE2)	NLMFE3 (NLMFE4)	NLMFE5 (NLMFE6)	NLMFE7 (NLMFE8)	NSSFE1 (NSSFE2)	NSSFE3 (NSSFE4)	NSSFE5 (NSSFE6)
CBCHG	Changes to source-bed head for transient leakage and to indicators for additional changes.		x (x)		x (x)					
COCHG	Changes to point source and sink, areally distributed source and sink, source-bed head for steady-vertical leakage, specified flux or head on Cauchy-type boundary, and specified head, and to indicators for additional changes.	x (x)	x (x)	x (x)	x (x)	x (x)	x (x)	x (x)	x (x)	x (x)
GNCHG	Changes to boundary and external head on nonlinear, head-dependent (Cauchy-type) boundary and to indicators for additional changes.						x (x)			
NXTPD	Initial values of indicators for time varying stresses and boundary conditions and time-step sizes.	x (x)	x (x)	x (x)	x (x)	x (x)	x (x)	x (x)	x (x)	x (x)
VNCHG	Changes to controlling head or altitude for nonlinear steady vertical leakage and to indicators for additional changes.					x (x)				

EXTRAP for linear versions of MODFE, and in subroutine WTXTRP for nonlinear versions, although the boundary condition is linear.

Nodes used to simulate specified-head boundaries are identified by an indexing scheme that distinguishes them from other nodes in the finite-element mesh. Because finite-element equations are neither formed nor solved at specified-head nodes (see Cooley (1992), section "Definition of Matrix Equation"), the indexing provides a mechanism for decreasing the order of the finite-element matrix  $A$  in equation (254) in Cooley (1992), and for numbering specified-head boundaries and equations to be solved. A description of the indexing of specified-head nodes and nodes where matrix equations are formed is given in the section "Reordering of Finite-Element Equations for Solution."

### Specified-Flux Boundaries

The volumetric flow rate per unit length, or unit discharge,  $q_B$ , [length<sup>2</sup>/time] across the specified-flux boundary is input by boundary-condition zone in subroutine DATIN. Two options are given in the input instructions for MODFE (Torak, 1993) that determine the program variables used to represent  $q_B$  for input. Option 1: If each element side within the boundary zone has the same unit discharge, then  $q_B$  represents a zone value and is input as program variable QBZ. After inputting the boundary-side number,  $J$ , and the node numbers KQB( $J$ ) and LQB( $J$ ) that define the element side on the boundary, QBZ is stored in the

program vector QBND( $J$ ) for side  $J$ . Option 2: If each element side within the boundary zone has a unique unit discharge, then unique  $q_B$  values for each boundary side are input as the program variable QBND( $J$ ), for each side  $J$ , along with values for  $J$ , KQB( $J$ ), and LQB( $J$ ). An indicator variable, IZIN, is input for each zone to identify the input option that is used. Details about preparing inputs to boundary-condition zones are given in Torak (1993).

The specified-flux term given by equation (50) in Cooley (1992) for Cartesian coordinates, and equation (229) in Cooley (1992) for axisymmetric cylindrical coordinates, is computed in subroutine FMCO. Values of QBND( $J$ ) for side  $J$  are assigned to the program variable QB so that QBND( $J$ ) also can store the  $\alpha$  term associated with the head-dependent (Cauchy-type) boundary (discussed in the following section). The unit discharge,  $Q_B$ , is multiplied by one-half the length of the element side on the boundary (program variable DIST) to form the specified-flux term in Cartesian coordinates for both nodes on the boundary. The value of DIST is stored for each node as the program variables TMPA and TMPB. For axisymmetric cylindrical coordinates, TMPA and TMPB are multiplied (weighted) by the average radial distance corresponding to DIST for each node (see equation (229) in Cooley, 1992). The lengths TMPA and TMPB are stored in the program vectors CFDK( $J$ ) and CFDL( $J$ ) for nodes KQB( $J$ ) and LQB( $J$ ), respectively, on boundary side  $J$ . Values of CFDK( $J$ ) and CFDL( $J$ ) are multiplied by the  $\alpha$  term if a head-dependent (Cauchy-type) boundary also is simulated along side  $J$ .



Table 11.—Subroutines that update hydraulic head and aquifer thickness

Subroutine	Description of computations	Versions of MODFE that require subroutine								
		LMFE1 (LMFE2)	LMFE3 (LMFE4)	NLMFE1 (NLMFE2)	NLMFE3 (NLMFE4)	NLMFE5 (NLMFE6)	NLMFE7 (NLMFE8)	NSSF1 (NSSF2)	NSSF3 (NSSF4)	NSSF5 (NSSF6)
HCALC	Updates hydraulic head to average value for time step or to new value for iteration level by using head changes from matrix solver.	x (x)	x (x)					x (x)	x (x)	x (x)
HCALWT	Updates hydraulic head to average value for predictor or corrector steps by using head changes obtained from matrix solver.			x (x)	x (x)	x (x)	x (x)			
SWBDMP	Damps computed head change obtained during water-table iteration.							x (x)	x (x)	x (x)
SWTHK	Updates aquifer thickness and transmissivity after water-table iteration.							x (x)	x (x)	x (x)

Table 12.—Subroutines that compute mass balance

Subroutine	Terms computed	Versions of MODFE that require subroutine								
		LMFE1 (LMFE2)	LMFE3 (LMFE4)	NLMFE1 (NLMFE2)	NLMFE3 (NLMFE4)	NLMFE5 (NLMFE6)	NLMFE7 (NLMFE8)	NSSF1 (NSSF2)	NSSF3 (NSSF4)	NSSF5 (NSSF6)
GNBAL	Volumetric rates for time step and total volumes during simulation for nonlinear head-dependent (Cauchy-type) boundaries and nonlinear point sinks. Augments flow rate at specified-head boundaries and flow imbalance of rates.						x (x)			
GNBLSS	Volumetric rates for time step and total volumes during simulation for nonlinear head-dependent (Cauchy-type) boundaries and nonlinear point sinks. Augments flow rate at specified-head boundaries and flow imbalance of rates.									x (x)
MASBAL	Volumetric rates for time step or for steady-state simulation for: accumulation of water in aquifer storage, steady vertical leakage, flow across specified flux and head-dependent (Cauchy-type) boundaries and at specified-head boundaries, and flow imbalance.	x (x)						x (x)	x (x)	x (x)
MBALCB	Same as MASBAL, and volumetric flow rate from transient leakage.		x (x)							
MBALWT	Same as MASBAL, except storage effects from conversion between confined and unconfined conditions are computed for accumulation of water in aquifer storage.			x (x)		x (x)	x (x)			
MBWTCS	Combined terms from MBALCB and MBALWT.				x (x)					
VNBAL	Volumetric rates for time step and total volumes during simulation for nonlinear steady vertical leakage. Augments flow rate at specified-head boundaries and flow imbalance of rates.					x (x)				
VNBLSS	Volumetric rates for time step and total volumes during simulation for nonlinear steady vertical leakage. Augments flow rate at specified-head boundaries and flow imbalance of rates.								x (x)	



The specified-flux term appearing in equations (50) and (229) in Cooley (1992) is added to the B vector of equations (1) and (2) in subroutine FMCO. These terms are computed as the product of  $Q_B$  and either TMPA or TMPB, and are added to the appropriate storage location in program vector Q.

Values of known fluxes, stored in the vector Q, are assembled into finite-element matrix equations (1) and (2) in subroutine FMEQ. Nodal fluxes in the Q vector are added into the program vector B, which stores the right side of the matrix equations. Because the specified flux is a known condition, the equation formulation only involves the right side of these equations.

### Head-Dependent (Cauchy-Type) Flux Boundaries

The computational steps of input, coefficient formation, and matrix-equation assembly for head-dependent (Cauchy-type) boundaries occur in the same subroutines as, and nearly concurrently with, the specified-flux terms described in the previous section. Inputs of the  $\alpha$  term and boundary head,  $H_B$ , defining the head-dependent (Cauchy-type) boundary (Cooley, 1992; Torak, 1993) are made in subroutine DATIN along with inputs for specified-flux conditions. The boundary, or external head,  $H_B$ , is represented by the program variables HK(J) and HL(J), respectively, for nodes k and l defining boundary-side J. Node numbers for the boundary side are represented by the program variables KQB(J) and LQB(J) for nodes k and l, respectively, on the boundary-side J. Two options of inputting  $\alpha$  are available. Zone values for  $\alpha$  are represented by the program variable ALPHZ, and values for each boundary side in the zone are represented by the program variable ALPH(J).

Terms that represent head-dependent (Cauchy-type) boundaries in the V matrix and B vector of equations (1) and (2) are formed in subroutine FMCO. These terms are expressed as the  $\alpha$  terms of equations (37), (50), and (229) in Cooley (1992). The length of the boundary side that multiplies the  $\alpha$  term in these equations is represented by program variables TMPA and TMPB. These variables contain the same value (equal to one-half the length of the boundary side) for simulations that use Cartesian coordinates, and are weighted for axisymmetric cylindrical coordinates, as described in the previous section. The head-dependent (Cauchy-type) terms given by the above equations are computed and stored by the program vectors CFDK(J) and CFDL(J) for nodes KQB(J) and LQB(J), respectively, on boundary-side J. However, if the  $\alpha$  term is zero, indicating that only the specified-flux condition is present at the boundary, then CFDK(J) and CFDL(J), respectively, store the lengths that

were computed as TMPA and TMPB. The node number stored in KQB(J) is converted to a negative value as an indicator to exclude this element side from other computational steps that involve head-dependent (Cauchy-type) boundaries.

The terms stored in CFDK(J) and CFDL(J) for head-dependent (Cauchy-type) boundaries are used in subroutine FMEQ to assemble finite-element matrix equation (254) in Cooley (1992). The terms CFDK(J) and CFDL(J) are added to the appropriate locations in program vector A for storing the main diagonal of the A matrix for nodes on the boundary. The right side of the matrix equation for nodes on the boundary contains the product of the head difference,  $HK(I) - H(K)$  or  $HL(I) - H(L)$ , and the head-dependent (Cauchy-type) boundary terms, CFDK(J) or CFDL(J). These products are added to storage locations of program vector B, which stores the right side of the matrix equation during the matrix-equation assembly.

## Programming Details of Nonlinear Hydrologic Terms

Programming details about the mathematical processes and computations related to input, formation, and assembly of matrix-equation components for terms contained in nonlinear versions of MODFE (tables 2 and 3) are given in the following sections. The nonlinear versions differ from linear versions (table 1) in that each nonlinear version simulates hydrologic processes that require reformulation of matrix-equation components during simulation. Reformulation is required for conditions where the hydraulic head in the aquifer changes position with reference to a controlling head or altitude associated with the hydrologic process such that different equations are needed to define the process mathematically. An example of a nonlinear condition is leakage from a riverbed when the water level in the aquifer fluctuates above and below the altitude of the bottom of the riverbed sediments during simulation. The changes in aquifer head with regard to the reference altitude (the altitude of the bottom of the riverbed sediments) require different equations for defining the volumetric flow rate between the aquifer and the river. The possible changes to the equation formation of nonlinear hydrologic processes are taken into account by the nonlinear versions of MODFE.

During the following descriptions of programming details for nonlinear hydrologic terms in MODFE, frequent reference is made to the finite-element matrix equations for steady- and nonsteady-state con-

ditions; thus, these equations are restated here. For nonsteady-state conditions, two matrix equations are used that correspond to the predictor and corrector steps. For the predictor step, equation (76) in Cooley (1992) is given as

$$\left( \frac{\underline{C}}{\left( \frac{2}{3} \right) \Delta t_{n+1}} + \underline{G}_n + \underline{V} \right) \delta^* = \underline{B} - (\underline{G}_n + \underline{V}) \hat{h}_n \quad (3)$$

For the corrector step, equation (80) in Cooley (1992) is given as

$$\left( \frac{\underline{C}}{\left( \frac{2}{3} \right) \Delta t_{n+1}} + \underline{G}^* + \underline{V} \right) \delta = \underline{B} - (\underline{G}^* + \underline{V}) \hat{h}_n \quad (4)$$

For nonlinear, steady-state conditions, finite-element matrix equation (234) in Cooley (1992) is given as

$$\underline{A}_i \delta_i = r_i \quad (5)$$

where similar terms in equations (1) and (2) have been described in the section "Programming Details of Linear Hydrologic Terms." The new terms in the above equations are derived from the following equations in Cooley (1992):  $\delta^*$ , equation (79);  $\underline{G}^*$ , equation (81);  $\underline{G}^*$ , equation (82);  $\delta_i$ , equation (237); and  $r_i$ , equation (238). The head change for the corrector step,  $\delta$ , is described in Cooley (1992) in the development following equation (80).

## Water-Table (Unconfined) Conditions

Inputs for water-table conditions are made to MODFE in subroutines FMCO, SWFMCO, and WTFMCO. Hydraulic conductivity in the x and y directions are input in subroutine FMCO by hydraulic-property zone as program variables XTR and YTR, respectively. Nodal values of aquifer thickness and altitude of the top of the aquifer (either land surface or the altitude of the bottom of an overlying confining bed) are input to subroutine WTFMCO for nonsteady-state conditions, and are input to subroutine SWFMCO for steady-state conditions. Aquifer thickness is represented by the program vector THK; the altitude of the top of the aquifer is represented by program vector TOP.

The specific yield of the unconfined aquifer is input to subroutine WTFMCO by hydraulic-property zone, and is represented by the program variable SY. Input of the artesian storage coefficient may be required in addition to specific yield when simulating unconfined conditions that can convert to confined conditions during simulation. Details about inputs that are required for simulating conversion between confined and unconfined conditions are given in the appropriate sections of Torak (1993). Programming details about forming storage-coefficient (capacitance) and transmissivity terms, and about updating aquifer thickness for unconfined conditions are given in the following sections.

### Storage Coefficient (Capacitance)

The storage-coefficient, or capacitance, terms for the C matrix in equations (3) and (4) are computed in subroutine WTFMCO according to equations (36) and (47) in Cooley (1992). Although these equations describe the formulation for confined conditions, the mathematical processes indicated by these equations are applied to unconfined conditions by replacing the artesian storage coefficient, S, with the specific yield, SY. The capacitance term  $c_{ii}^e$  of equation (36) in Cooley (1992) represents the contribution from element e to the C matrix for node i. These element contributions are computed as the program variable TESY by multiplying the specific yield (SY) by one-third the element area, which is stored in the program vector AR. Values of TESY are identical for each node in the element, and are summed for all elements connected to node i according to equation (47) in Cooley (1992) by adding their value to the appropriate storage location in program vector ASY. This summation yields the main-diagonal term in the C matrix for capacitance.

The capacitance term is assembled into finite-element matrix equations (3) and (4) in subroutines FMEPWT and FMECWT, respectively. The main-diagonal terms that are stored in program vector ASY are assigned to the storage locations A(NC) of program vector A. Values in the A(NC) locations for each equation are multiplied by  $3/[2\Delta t_{n+1}]$ , represented by program variable DTM, to effect the division by  $(2/3)\Delta t_{n+1}$  indicated in the matrix equations. Program variable TMPA is used to store the product of A(NC) and DTM. Values of TMPA are added to the storage locations in program vector A, A(NME), that represent main diagonals of the C, V, and G matrices. The formulation of the capacitance term for vector  $\underline{B}$  of the corrector step is performed by multiplying TMPA by the average predicted head change for the time step,  $\delta^*$ , represented by program variable DH.

### Transmissivity

Computations for the transmissivity terms contained in the  $G$  matrices of equations (3)–(5) are performed in subroutines FMCO, FMECWT, FMEPWT, SWFMCO, and SWTHK. Element contributions to the hydraulic-conductivity terms given by equation (72) in Cooley (1992) are computed in subroutine FMCO as the program variables TFL(1), TFL(2), and TFL(3). These computations are identical to the computations described previously in the section “Transmissivity,” as the same subroutine is used to form transmissivity terms for linear and nonlinear conditions. The element contributions (TFL terms) to each node are summed in subroutine FMCO to form the hydraulic-conductivity terms,  $D_{ij}$ , given by equation (75) in Cooley (1992). The  $D_{ij}$  terms are represented by storage locations in the program vector A.

The transmissivity terms of the  $G$  matrices in equations (3)–(5) are formed in subroutines FMECWT, FMEPWT, SWFMCO, and SWTHK. For nonsteady-state simulations, subroutine FMEPWT is used for the predictor step and subroutine FMECWT is used for the corrector step. For steady-state simulations, the transmissivity terms are computed initially in subroutine SWFMCO and are updated on each iteration in subroutine SWTHK. Details of these computations are given in the following paragraphs.

The formation of transmissivity terms for the predictor step, equation (3), is made in subroutine FMEPWT. Off-diagonal transmissivity terms for the  $G_n$  matrix of this equation are computed according to equation (77) in Cooley (1992), and are represented by the program variable TMPA. Aquifer thickness in these computations is represented by program variables THKI and THKL for the two nodes that are linked by the transmissivity term. The hydraulic-conductivity term,  $D_{ij}$ , of equation (77) in Cooley (1992) is represented by the location A(ND+J) of program vector A.

The transmissivity coefficients computed as TMPA are assembled into the  $G_n$  matrix of finite-element equation (3) in subroutine FMEPWT. The off-diagonal terms (values of TMPA) are assigned to the storage locations A(NME+J) of program vector A. The main diagonal of the  $G_n$  matrix is formed according to equation (77) in Cooley (1992) by adding the negative values of TMPA to the appropriate storage location in program vector AD. After matrix components to all equations are formed, the main diagonal is stored in the A vector, which is identified in this subroutine by the location A(NME).

Formation of transmissivity terms for corrector-step equation (5) is made in subroutine FMECWT. Off-diagonal terms for  $\tilde{G}^*$  and  $\bar{G}^*$  in this equation are computed, respectively, according to equations (81)

and (82) in Cooley (1992). In these computations, aquifer thickness at the beginning of the time step and average thickness changes from the predictor step are used to multiply the hydraulic-conductivity terms,  $D_{ij}$ , to obtain the off-diagonal transmissivity terms. Aquifer thickness is represented by program variables THKI and THKL, and thickness changes from the predictor step,  $\delta^*$  of equation (81) in Cooley (1992), are represented by program variables DTK(I) and DTK(L) for the two nodes that are linked by the transmissivity term. The hydraulic-conductivity term,  $D_{ij}$  in equation (75) in Cooley (1992), is represented by the location A(ND+J) in program vector A. The off-diagonal term,  $\tilde{G}_{ij}^*$ , is computed as program variable TMPA.

Computations for the off-diagonal term,  $\tilde{G}_{ij}^*$ , are combined with computations contained in  $\bar{B}$  that involve  $\bar{G}_{ij}^*$ , to yield the term for the right side of the corrector equation (see development leading to equation (80) in Cooley, 1992). Terms for the right side of the corrector equation are stored in program vector B. Computations of the right-side terms are identified by program statements that contain the multiplier, (1/16), which is represented by program variable C3.

The main-diagonal terms for  $\tilde{G}_{ij}^*$  are computed according to equation (43) in Cooley (1992) by summing the negative values of the off-diagonal terms into storage locations in the program vector AD. These computations are identified by statements that contain the program vector AD and either TMPA or an explicit formulation of the off-diagonal term involving aquifer thickness (THKI and THKL) and predicted thickness changes (DTK(I) and DTK(L)). After these computations have been performed for all matrix equations, the main-diagonal terms are stored in the locations A(NME) of program vector A.

For steady-state conditions, the transmissivity terms are computed in subroutines SWFMCO and SWTHK. Initial values for the off-diagonal transmissivity terms, given by equation (74) in Cooley (1992), are computed in subroutine SWFMCO after inputs of aquifer thickness (THK) and altitude of the aquifer top (TOP) are made. The hydraulic-conductivity terms,  $D_{ij}$ , stored in program vector A are multiplied by the average aquifer thickness along an element side to yield the off-diagonal transmissivity term. Aquifer thickness for both nodes along the element side are represented by storage locations THK(I) and THK(K), and the hydraulic-conductivity and transmissivity terms are represented by the storage location A(ND+J).

Transmissivity terms are computed during iteration in subroutine SWTHK. Updated values of aquifer thickness are used to multiply the hydraulic-conductivity term, A(ND+J), to obtain the off-

diagonal transmissivity term given in equation (244) in Cooley (1992). Aquifer thickness for iteration level  $l$  is represented by program variables THKI and THKK for the two nodes along an element side corresponding to the transmissivity term. The off-diagonal transmissivity term is stored in location A(ND+J) of program vector A.

Transmissivity terms are assembled into the finite-element matrix equation (5) in subroutine FMEQ. Programming details for adding transmissivity terms to the A vector to form the equations for nonlinear steady-state conditions are identical to those presented for the linear case. (See section "Transmissivity" under "Programming Details of Linear Hydrologic Terms.")

#### Updates to Aquifer Thickness

Aquifer thickness is updated at the end of the current time step for nonsteady-state conditions in subroutine XTRPWT, and at end of the current iteration level for steady-state conditions in subroutine SWTHK. Note that for nonsteady-state conditions, aquifer thicknesses stored in program vector THK are not updated to provide estimates of transmissivity terms for the corrector step (discussed in the previous section). Instead, predicted thickness changes, stored in program vector DTK, are used to augment aquifer thickness, stored in THK, when computing transmissivity terms for the corrector step.

Updates to aquifer-thickness values stored in program vector THK are made in subroutine XTRPWT according to equation (85) in Cooley (1992). These updates are made at each node following extrapolation of hydraulic head from average values for the time step to values at the end of the time step. Aquifer heads at the beginning and end of the time step are represented by program variables HO and the vector H, respectively. These values are compared with the altitude of the top of the aquifer, stored in program variable TOP, to determine if the total head change for the time step can be used to update aquifer thickness. The total head change is computed from the average head change, stored in program variable DHC, as  $1.5 \times DHC$ . If conversion between confined and unconfined conditions occurred at a node, then the thickness change is limited to the head change that occurred within the unsaturated part of the aquifer (fig. 13).

Aquifer thickness is updated during each water-table iteration in subroutine SWTHK by computations that are similar to those described previously for subroutine XTRPWT (fig. 13). The initial aquifer head, HO, and the head after the water-table iteration, H(I), are compared with TOP(I) to determine if

the total head change during iteration can be used to update aquifer thickness. The head change is represented by program vector B.

### Conversion Between Confined- and Unconfined-Aquifer Conditions

Computations for conversion between confined and unconfined aquifer conditions are performed automatically in all nonlinear versions of MODFE. In nonsteady-state versions (table 2), subroutines FMECWT and FMEQWT perform the appropriate comparisons of aquifer head at the beginning and end of the time step with the altitude of the top of the aquifer (or base of an overlying confining bed) to determine if conversion has occurred at a node. In steady-state versions (table 3), these comparisons occur in subroutine SWTHK by using the computed head for the current iteration.

Aquifer head, H(I), is compared to the altitude of the top of the aquifer, TOP(I), for each node I in predictor equation (3) in subroutine FMPEWT. Based on these comparisons, either the aquifer storage coefficient or specific yield is used to form the capacitance term for the C matrix in equation (3). The storage coefficient is represented by the locations A(NC) in the A vector, and the specific yield is represented by the vector ASY. After comparing H(I) with TOP(I), the appropriate storage term is assigned to the A(NC) locations, which are used to form the matrix equations.

For corrector equation (4), aquifer heads at the beginning and end of the time step are compared with TOP in subroutine FMECWT to determine if conversion between confined and unconfined aquifer conditions has occurred. The predicted aquifer head at the end of the time step is represented by program variable HP, and head at the beginning of the time step is represented by program variable HO. After comparing HP and HO with TOP(I), a predicted change in aquifer thickness, DTK(I), is computed for node I. Values for DTK(I) are assigned as either the total predicted head change for the time step, computed as program variable DHP, or the part of the predicted head change that occurred below TOP(I) during the conversion (fig. 14). Predicted thickness changes are multiplied by 2/3 to obtain average thickness changes for the time step. These values and the thickness values from the beginning of the time step, program vector THK, are used to compute transmissivity coefficients for the corrector step.

For confined to unconfined conversions, the predicted head, HP, is revised in subroutine FMECWT according to equation (100) in Cooley (1992) to account for inaccuracies that are associated with computing

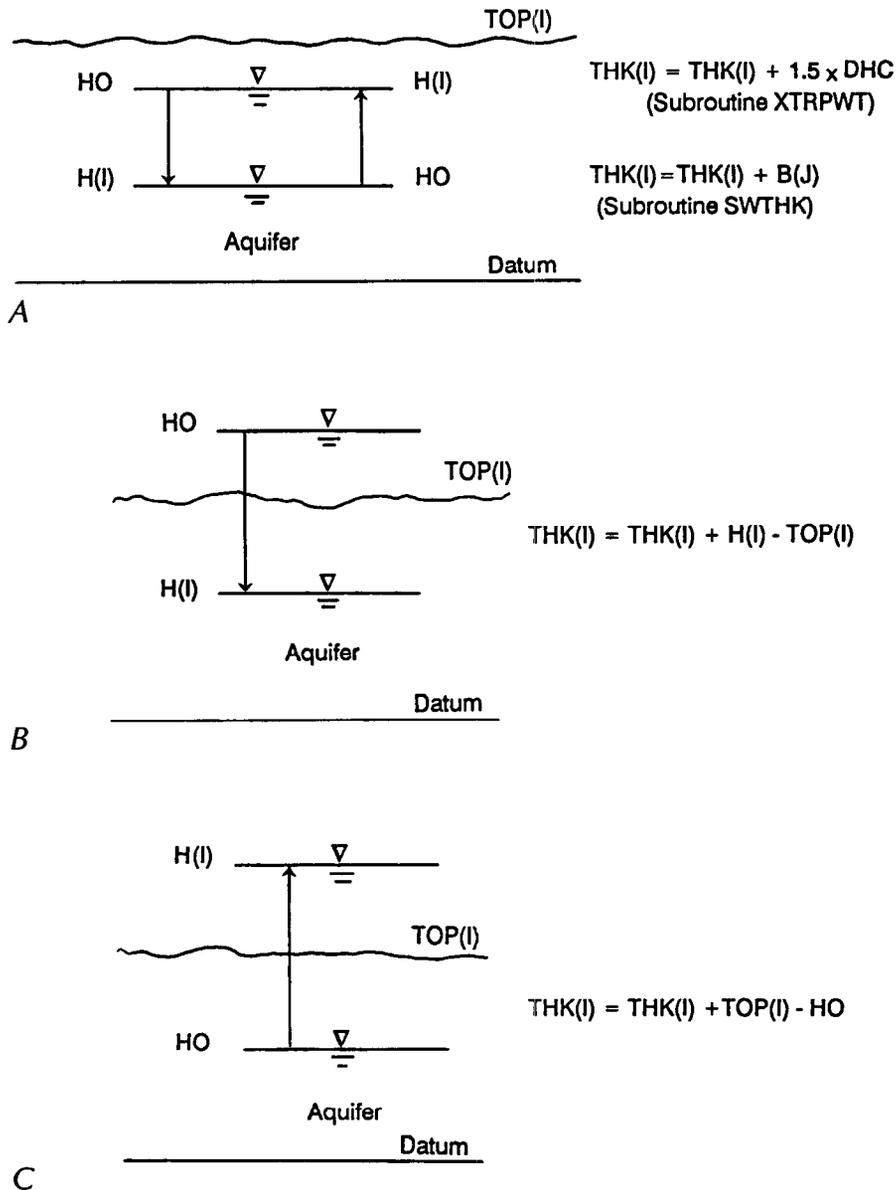


Figure 13.—Positions of aquifer head, HO, at beginning of time step, and H(I), at end of time step, and computations in MODular Finite-Element model (MODFE) for updating aquifer thicknesses, THK(I), at node I for (A) water-table (unconfined) conditions without conversion, (B) conversion from confined to unconfined conditions, and (C) conversion from unconfined to confined conditions.

HP on the predictor step by using only the artesian storage coefficient (see development leading to equation (100) in Cooley 1992). The revised predicted head, HP, is used to recompute the predicted thickness change, DTK(I), predicted head change, DHP, and the average head change, which is represented by program vector DH.

Changes to the corrector equation (4) caused by conversion between confined and unconfined aquifer conditions are made in subroutine FMECWT. The

capacitance term in the C matrix of equation (4), stored in location A(NC) in program vector A, is assigned the value of specific yield, stored in program vector ASY, for the node that converts to unconfined conditions. The right side of equation (4) is modified to contain the capacitance term shown in equation (98) in Cooley (1992). These computations involve program vector B, which stores the right side of equation (4), and occur as program statements  $B(J) = B(J) + (ASY(I) - A(NC)) \times (TOP(I) - HO)/DT$  for confined

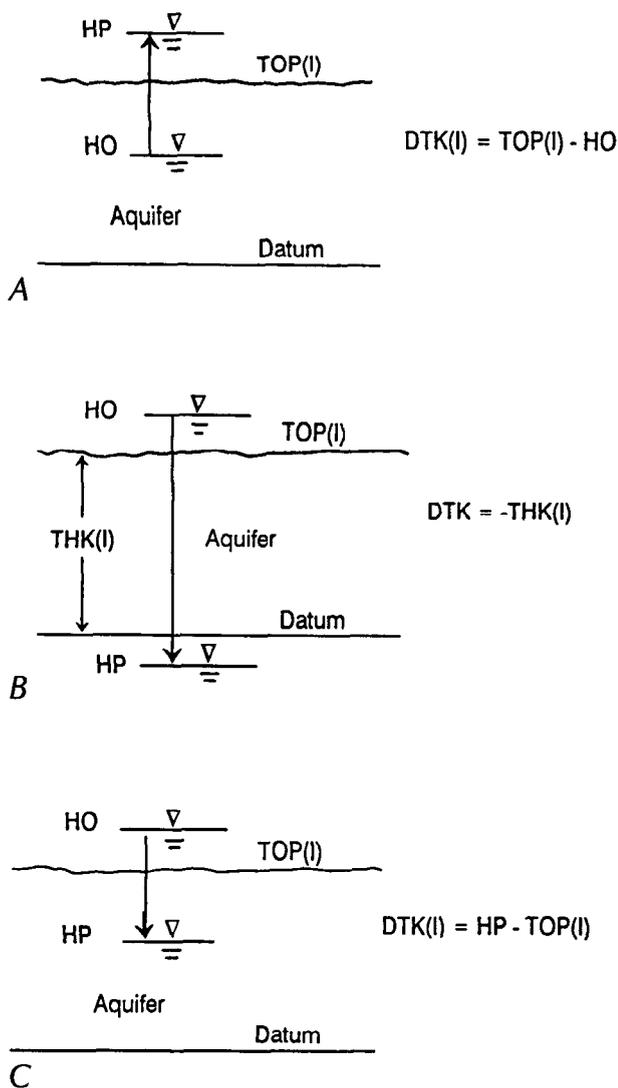


Figure 14.—Positions of aquifer heads, HO and HP, and computation of predicted thickness change, DTK(I), at node I for conversion from (A) unconfined to confined conditions, and (B) and (C) confined to unconfined conditions.

to unconfined conversions, and  $B(J) = B(J) + (A(NC) - ASY(I)) \times DTK(I)/DT$  for unconfined to confined conversions, where DT is the time-step size.

Hydrologic conditions that cause unpredicted conversions or nonconversions are evaluated in subroutine WTCCHK. Predicted heads at the end of the time step are computed as program variable HP by using average head changes from the predictor step, stored as program vector DH. After comparing HP and heads at the beginning of the time step with the altitude TOP, HP is recomputed by using the head changes from the corrector step, which are stored in program vector B. If new values of HP indicate conversion, then either an unpredicted conversion or

nonconversion had occurred during the corrector step, and the indicator variable ISC is set to 1. The indicator is evaluated by the main programs of the nonlinear (nonsteady-state) versions of MODFE (table 2); for  $ISC=1$ , the corrector step is repeated using thickness and transmissivity updates from the previous corrector as predicted values.

Computations for conversion during steady-state simulations are performed in subroutine SWTHK and involve computing transmissivity terms with updated thickness values from the current water-table iteration. At nodes that experience conversion, the thickness change that is added to program vector THK is that part of the head change that occurs below the altitude of the top of the aquifer or base of the overlying confining bed (fig. 13A,B). Programming details of updating aquifer thickness in this subroutine are given in the previous section.

### Aquifer Drying and Resaturation

Computations for aquifer drying and resaturation are incorporated into subroutines that form coefficients and that form and solve finite-element matrix equations. A node is considered dry when the thickness, stored in program vector THK, is computed to be zero or negative. Resaturation occurs when the thickness at a dry node is computed to be greater than zero. Checks for zero or negative thickness values are placed in subroutines FMECWT and FMEPWT for nonsteady-state conditions, equations (3) and (4), and in subroutine SWTHK for steady-state conditions, equation (5). Aquifer thickness is updated after each time step and iteration level in subroutines XTRPWT and SWTHK, respectively (see discussion in section "Updates to Aquifer Thickness"). Hydrologic implications of a negative aquifer thickness at a node are explained in the section "Drying and Resaturation of Aquifer Material" in Torak (1993).

Computations for transmissivity coefficients use temporary-valued variables for aquifer thickness and for changes in thickness to account for aquifer drying at a node. Values for aquifer thickness, stored in program vector THK, are assigned to program variables THKI and THKL for the two nodes where a transmissivity coefficient is computed. If either or both nodes is dry (that is, if the corresponding value in THK is less than or equal to zero), then the corresponding variables, THKI, THKL, or both, are set to zero. For corrector-step computations in subroutine FMECWT, different values for the predicted thickness change, stored in program vector DTK, are assigned to nodes predicted to go dry depending on whether the node was saturated or dry at the beginning of the time step. Dry nodes that are predicted to

stay dry have zero values assigned to the appropriate location in vector DTK, while nondry nodes that are predicted to go dry have values of  $-THK$  assigned to vector DTK. The computations for transmissivity coefficients proceeds as described in the section "Transmissivity" for nonlinear hydrologic terms by using values assigned to program variables THKI and THKL, and to program vector DTK.

Negative fluxes (indicating discharge) that are used to form the right side of equation (4) for the corrector step and (5) for steady-state conditions are decreased by half of their current value at nodes that are predicted to go dry. For the corrector step, a zero or negative value for the predicted aquifer thickness at a node (program variable THKP) is used to determine the drying condition for changing the negative fluxes. Values of THKP are computed as the sum of aquifer thickness,  $THK(I)$ , and predicted aquifer thickness,  $DTK(I)$ , for each node  $I$ . For steady-state conditions, zero or negative values for the updated aquifer thickness, stored in program vector THK, are used to determine dry nodes and subsequent decreases to the known fluxes. Fluxes for the right side of equations (4) and (5) are stored in location  $Q(I)$  in program vector  $Q$  for node  $I$ .

Decreases in negative fluxes at dry nodes are accounted for in the water-balance summary by decreasing the term  $WQO$  used to represent point sinks. The decrease to  $WQO$  is completely arbitrary, as values for individual known fluxes are not stored in  $MODFE$ , and the net negative flux at a dry node may not be attributed to point sinks. Therefore, a message is printed out from subroutines  $FMECWT$  and  $SWTHK$  giving the value of the decrease in net negative flux at the dry node so the user can adjust the appropriate terms in the water-balance summary.

### Head-Dependent (Cauchy-Type) Flux

Inputs for nonlinear head-dependent (Cauchy-type) boundaries are made in subroutines  $GNINIT$  and  $GNFMCO$ . Values for the number of element sides and the number of boundary-condition zones that represent nonlinear head-dependent (Cauchy-type) boundaries are input in subroutine  $GNINIT$  as program variables  $NBNC$  and  $NLCZ$ , respectively. These values are used to allocate computer storage for the boundary condition. Inputs for defining boundary-condition zones and for selecting an option for entering values for the  $\alpha_r$  term of equation (153) in Cooley (1992) are made in subroutine  $GNFMCO$ . These inputs consist of the zone number, the number of sides in the zone, and an indicator variable for the input of  $\alpha_r$ , which are represented, respectively, by program variables  $KZ$ ,  $NOS$ , and  $IZIN$ . Formats and descrip-

tions of these inputs are given in the section "Input Instructions" of Torak (1993). Other inputs for nonlinear head-dependent (Cauchy-type) boundaries that are made in subroutine  $GNFMCO$  are the number of the element side on the boundary (program variable  $J$ ), node numbers defining the element side (program variables  $KR(J)$  and  $LR(J)$ ), boundary or external heads (program variables  $HRK(J)$  and  $HRL(J)$ ), controlling heads or altitudes (program variables  $ZRK(J)$  and  $ZRL(J)$ ), and the  $\alpha_r$  term that controls the flux. The  $\alpha_r$  term is input either as program variable  $GCZ$  for the boundary-condition zone, or as  $GC(J)$  for each side  $J$ .

Formulation of the coefficient  $C_r$  of equation (155) in Cooley (1992), for nonlinear, head-dependent (Cauchy-type) boundaries occurs in subroutine  $GNFMCO$ . The  $\alpha_r$  term, stored in program vector  $GC$ , is multiplied by half the length of the element side on the boundary to complete the formulation. Program variable  $DIST$  is used to compute and store the length term used in these computations.

Components of matrix equations (3)–(5) for nonlinear, head-dependent (Cauchy-type) boundaries are assembled in subroutines  $GNPRED$  and  $GNCORR$  according to the formulations (with the appropriate substitutions) given by equations (124)–(127) in Cooley (1992) for areal, head-dependent leakage (see development following equation (155) in Cooley (1992)). Each formulation of the nonlinear boundary condition is identified in these subroutines by a **COMMENT** statement containing a case number, 1–4, which corresponds to equations (124)–(127), respectively. Comparisons are made between aquifer head and controlling head or altitude to identify the formulation, or case, that applies to a particular boundary side. For the predictor step and for steady-state conditions, aquifer head is represented by program vector  $H$  and controlling head or altitude is represented by program vector  $ZR$ . The boundary or external head is represented by program vector  $HR$ . Cases 1 and 4 are evaluated in subroutine  $GNPRED$  for forming coefficients for the predictor step and for steady-state conditions; all four cases are evaluated in subroutine  $GNCORR$  for the corrector step.

Computations for nonlinear, head-dependent (Cauchy-type) boundaries are simplified in subroutine  $GNCORR$  for forming coefficients for the corrector step by using the average aquifer head and average boundary or external head, and by representing heads, altitudes, and head differences by nonsubscripted variables. The average aquifer head,  $\bar{h}$ , and average boundary or external head,  $\bar{H}_r$ , are obtained by applying equation (63) in Cooley (1992) to  $\hat{h}$  and  $\hat{H}_r$ , and are represented by program vectors  $H$  and  $HR$ , respectively. Aquifer head at the beginning and end of

the time step is represented by program variables HO and HP, respectively. For each node I on the boundary, the controlling head or altitude is represented by program variable ZRI, and the boundary or external head is represented by program variable HRI. Head or altitude differences HRI-ZRI and HO-ZRI are represented by program variables DRZ and DHZ, respectively. The values for DRZ and DHZ, and the predicted head change for the time step, program variable DHP, are used to compute the  $\phi$  terms, given by equations (111) and (115) in Cooley (1992), which are used to proportion equations for the nonlinear boundary condition over the time step for cases in which two expressions for the boundary flux apply during the time step. (See development in Cooley (1992) leading to equations for the  $\phi$  terms.)

Terms for nonlinear head-dependent (Cauchy-type) boundaries are added into matrix equations (3)–(5) in subroutine GNPRED, for the predictor step and for steady-state conditions, and in subroutine GNCORR, for the corrector step. For each case, terms for the G matrix and the right side of matrix equations (3)–(5) are added to the appropriate locations in program vectors A and B. The main diagonal for the V matrix is represented by location A(N) in the A vector and the right side is represented by the location B(LE) in the B vector, where indexes N and LE correspond to the equation number that is being formed. Depending on the case, the  $C_r$  coefficient, represented by program variable GC(J), is added to the main diagonal of the V matrix, and the product of  $C_r$  and a head or altitude difference is added to the right side of the matrix equations. Also, the main diagonal and right-side component of the matrix equations is multiplied by the  $\phi$  terms for cases where two expressions are needed to represent the boundary flux during the time step (cases 2 and 3). The head or altitude difference is computed for the appropriate cases and stored as program variable TMPA, and the  $\phi$  terms are represented by program variables PHI and PHC. The product of TMPA and GC(J) is added to the storage location B(LE) to complete the formulation of terms for the right side of the matrix equations.

## Point Sinks

The computational steps of providing input, forming coefficients, and forming equations for nonlinear point sinks are performed in subroutines GNFMCO, GNPRED, and GNCORR in a manner similar to that described in the previous section for nonlinear, head-dependent (Cauchy-type) boundaries. The number of nonlinear point-sink boundaries is input to subroutine GNINIT as program variable NPNB, and is used to allocate storage locations for program vectors associ-

ated with this boundary condition. Inputs that define nodes as nonlinear point-sink boundaries are made in subroutine GNFMCO and consist of the number of the boundary, I, the node number on the boundary, KP(I), coefficient  $C_{pi}$  of equation (106) in Cooley (1992), and controlling head or altitude, ZP(I). The coefficient  $C_{pi}$  is input as program variable GCP and is stored in program vector GC by using storage locations IP that follow the locations assigned to similar terms for nonlinear, head-dependent (Cauchy-type) boundaries.

Because of the manner in which point-sink boundaries are defined, coefficients for matrix equations (3)–(5) are formulated completely by the values of  $C_p$  that were input in subroutine GNFMCO and are stored in program vector GC. Hence, additional computations to form coefficients for nonlinear point sinks are unnecessary.

Components of matrix equations (3)–(5) for nonlinear point sinks are assembled in subroutine GNPRED for the predictor step and for steady-state conditions, and in subroutine GNCORR for the corrector step. Aquifer head is compared with controlling head or altitude to determine the appropriate formulation of the boundary condition for the matrix equations. Four different cases are possible for representing nonlinear point-sink boundaries in the matrix equations. Three of the four cases are expressed by equations (108)–(110) in Cooley (1992); the fourth case requires no formulation (see development in Cooley (1992) leading to these equations). Depending on the case that is formulated, terms for nonlinear point-sink boundaries are added to the main diagonal of the V matrix and to the right side of the matrix equations. The main diagonal for the V matrix is represented by location A(N) in the A vector and the right side is represented by the location B(KE) in the B vector, where indexes N and KE correspond to the equation that is being formed.

Matrix-equation components of nonlinear point sinks for case 1 are formulated in subroutine GNPRED for the predictor step and for steady-state conditions. (Cases 2 and 3 are not applicable to the predictor step or to steady-state conditions and case 4 requires no formulation.) The  $C_p$  coefficient, represented by program vector GC, is added to the main diagonal of the G matrix, and the product of  $C_p$  and the head or altitude difference, ZP(I)–H(K), is added to the right side of the matrix equations, where the indexes define node K on boundary side I.

Matrix-equation components to nonlinear point sinks for cases 1–3 are formulated in subroutine GNCORR for the corrector step. Computations are simplified from those given by equations (108)–(110) in Cooley (1992) by using the average aquifer head for

case 1 and by using unsubscripted variables to represent aquifer head and head or altitude differences for cases 2 and 3. The average aquifer head is represented by program vector H, and heads at the beginning and end of the time step are represented, respectively, by program variables HO and HP. The controlling head or altitude, ZP, is represented by program variable ZPI, and the difference, ZPI-HO, is represented by program variable DHZ. These values are used to compute the  $\phi$  terms according to equations (111) and (115), which are used in formulating cases 2 and 3 to account for changes in the boundary-flux equations during the time step. The  $\phi$  terms are represented as program variables PHI and PHC. They multiply the main-diagonal term, GP(IP), for case 3 and the right-side terms for cases 2 and 3. Head or altitude differences are computed for the appropriate cases and stored as the program variable TMPA. The product of TMPA and GC(IP) is added to the storage location B(KE) to complete the formulation of terms for the right side of the matrix equations for the three cases.

### Steady Vertical Leakage

Input of hydrologic information, coefficient formation, and matrix-equation assembly for nonlinear steady vertical leakage occur in subroutines VNINIT, VNFMCO, VNCORR, and VNPRED. The number of hydraulic-property zones for nonlinear steady vertical leakage is input to subroutine VNINIT as program variable NVNZ. This value is used in the other "VN" subroutines as a counter to limit computations and inputs that pertain to this boundary condition. Values for the steady vertical leakage terms  $R_a$  and  $R_e$  of equations (117) and (129), respectively, in Cooley (1992) are input by hydraulic-property zone in subroutine VNFMCO. Program variable VNCF is used to represent the "R" terms. Other inputs defining the zone consist of the zone number, the beginning element number for the zone, and the number of elements in the zone. These values are represented, respectively, by program variables L, NBE, and NO. The controlling head or altitude, given by  $H_a$  and  $z_e$  in the equations referenced above is input by node as program vector HS. Details about assigning values to these variables and establishing hydraulic-property zones are given in Torak (1993).

Computations of coefficients for steady vertical leakage that add into the matrix equations are performed in subroutine VNFMCO. The coefficients  $C_a$  and  $C_e$ , given by equations (119) and (132), respectively, in Cooley (1992), are computed for each element in the hydraulic-property zone. These terms represent the contribution to the V matrix of equations (3) and (4) for each node where an equation is

formed. The formulation of these terms is similar to the formulation described previously for linear steady vertical leakage. Values for each element that contribute to these coefficients are computed as the product of the R term (VNCF) and one-third of the element area (stored in program vector AR). The element contributions are computed as program variable EC and are summed by program vector E over all elements that are connected to a node. The summation is performed efficiently by computing the indexes NA, NB, and NC for the three nodes in an element and by adding EC to storage locations E(NA), E(NB), and E(NC). These computations are repeated for all elements in the hydraulic-property zone and for all zones to complete the formulation of coefficients.

Matrix-equation components for nonlinear steady vertical leakage are added to the appropriate locations in equations (3)–(5) in subroutines VNPRED for the predictor step and for steady-state conditions, and in subroutine VNCORR for the corrector step. The main diagonal of the G matrix in these equations is represented by the location A(NME) of program vector A, and the right side terms are represented by the location B(K) of program vector B. The indexes NME and K correspond to the equation that is being formed for a node. The main-diagonal and right-side components of nonlinear steady vertical leakage are added to these locations based on evaluation of the leakage case, or formulation, that applies at the node.

The type of leakage that is simulated and subsequent equation formulation is determined by evaluating the sign of coefficient values stored in program vector E; negative values for discharge only, and positive values for recharge and discharge (see appropriate sections of Torak (1993) for details). Computations are bypassed at a node for zero values stored in the corresponding location of program vector E. For discharge-only leakage, nine different formulations, or cases, are possible by comparing aquifer head with land-surface and controlling altitudes. These cases are given by eight equations, (135)–(140) and (142)–(143) in Cooley (1992); case 7 requires no formulation. For the recharge and discharge type of leakage, four cases similar to those developed for the point head-dependent sink functions arise by comparing aquifer head with the altitude of the top of the aquifer or of the base of the overlying confining layer. However, only cases 1 and 4 of either type of leakage are formulated in subroutine VNPRED; the remaining cases do not apply to equations for the predictor step or for steady-state conditions, and case 7 of the discharge-only type requires no formulation.

Matrix-equation components for nonlinear steady vertical leakage are assembled in subroutine VNPRED by assigning the coefficient stored in pro-

gram vector E to program variable EI. Values of EI are made positive for cases involving the discharge-only function. Based on comparisons of aquifer head at the beginning of the time step with the controlling head or altitude, HS, and with the altitude of the top of the aquifer (or base of the overlying confining bed), TOP, a head or altitude difference is computed, which multiplies EI to become the right-side term of the matrix equations. The head or altitude difference is represented by program variable TMPA. The leakage coefficient, EI, is added to the location A(NME) to form the main-diagonal component of the V matrix in equations (3) and (4).

Computations that assemble matrix-equation components in subroutine VNCORR are simplified by representing aquifer head, controlling altitudes, and head and altitude differences as unsubscripted variables. Formulations for some of the leakage cases are further simplified by computing average values over the time step for aquifer and controlling heads according to equation (63) in Cooley (1992), and by combining computations that involve the  $\phi$  terms and the right side of the matrix equations. Aquifer heads at the beginning and end of the time step are represented, respectively, by program variables HO and HP. Values for HP are computed by using the predicted head change, computed as program variable DHP. The controlling altitudes or heads that are stored in program vector HS are represented as program variables ZE for the discharge-only leakage, and as HA for recharge and discharge. The altitude of the top of the aquifer is represented as program variable ZT. Head or elevation differences are represented by the following program variables: DEH, for ZE-HO; DHT, for HO-ZT; DET, for ZE-ZT; and DAT, for HA-ZT. Frequent computations involving the  $\phi$  terms are represented by the following program variables: PHC, PHCF, PHE, PHI, and PHT. The head or altitude differences that multiply the leakage coefficient EI are represented by program variable TMPA. These values are modified by the  $\phi$  terms for some of the leakage cases. The product of TMPA and EI comprises the component for the right side of matrix equation (4) for nonlinear steady vertical leakage. Values of EI are similarly modified by the  $\phi$  terms and are added to the location A(NME) to form the main-diagonal component of the G matrix in the corrector-step equation.

## Computational Aspects of Simulation Features

Computational aspects of features to MODFE that enhance its ability to simulate ground-water flow are

provided in this section. Details are given about specific computations, program structures, and versions that pertain to simulating linear and nonlinear steady-state conditions, changing stresses and boundary conditions with time, and computing a water-balance summary to assist the user in applying MODFE to a particular aquifer problem.

### Steady-State Flow

Steady-state conditions in an aquifer are simulated by either linear or nonlinear versions of MODFE. Linear versions solve equation (2) for the head change,  $\delta$ , that occurs as a result of stresses and boundary conditions to the simulated region. The head changes are computed without providing updates to hydrologic terms in the A matrix or B vector. Nonlinear versions solve equation (5) for the head change or displacement,  $\delta_i$ , that occurs during iteration. The displacements occur not only as a result of stresses and boundary conditions that are acting upon the simulated region, but because of changes (nonlinearities) in hydrologic terms contained in the A matrix and r vector, which require updating after each iteration. Linear conditions of steady-state flow are simulated by using the same versions of MODFE that are used to simulate linear, nonsteady-state conditions. However, nonlinear steady-state flow is simulated by versions of MODFE that are structured specifically for simulating nonlinear, steady-state conditions (see section "Simulation Capabilities and Versions of MODFE"). Computational aspects of versions of MODFE that simulate linear and nonlinear steady-state conditions are described in the following sections.

#### Linear Conditions

Linear steady-state conditions are simulated by versions of MODFE shown in the structure diagrams as LMFE1-LMFE4 (figs. 1 and 2). Formulation of matrix equation (2) is performed by inputting a value of 1 for the indicator variable ISTD. The indicator variable is evaluated in subroutine DATIN prior to computing known head changes at specified-head boundaries. For steady-state conditions, known head changes represent the total change in head from the initial conditions. For nonsteady-state conditions, the head change represents the average change in head for the time step, where the total head change is multiplied by 2/3. Known head changes are represented by program vector DHB.

Although the steady-state solution is not dependent on initial conditions, these conditions are incorporated into the matrix equations so that the equations solve for head changes,  $\delta$ , rather than head. This decreases

the magnitude of the unknowns in the matrix equations, thus decreasing roundoff error in the solution (see section "Evaluation of Time Integral" in Cooley, 1992).

The value of the indicator variable, *ISTD*, is checked in the main programs of MODFE to bypass computations that extrapolate head changes for nonsteady-state conditions. Extrapolation is necessary for nonsteady-state conditions as the solution vector,  $\delta$ , in equation (1) represents average head changes during the time step. For steady-state conditions, the solution vector,  $\delta$ , represents total head change from the initial conditions; therefore, extrapolation is not required.

A time-step size of unity (=1) is required as input to MODFE for simulating linear steady-state conditions. Although the steady-state solution is not time dependent, the time-step size is used in computations that assemble matrix equations and compute a water-balance summary. A value of 1 for the time-step size eliminates division by zero and permits values that appear in the water-balance summary to represent rates and volumes for one time unit. Instructions for inputting the time-step size are given in the section "Input Instructions" in Torak (1993).

### Nonlinear Conditions

Nonlinear steady-state conditions are simulated by the versions of MODFE shown in figures 9–12. These versions provide solutions to matrix equation (5) for the displacement,  $\delta_l$ , or head change for iteration level *l*. Values for the displacement vector,  $\delta$ , are computed alternately with updating nonlinear terms in the matrix equation by an iteration process. Because all nonlinearities represented in MODFE involve either water-table (unconfined) conditions or the potential for water-table conditions to occur during simulation, the iteration process is termed water-table iteration in the following discussions. Computations for water-table iterations necessitated that steady-state versions be constructed separately from nonsteady-state versions of MODFE.

Inputs for controlling water-table iterations are made in subroutine SWINIT, and consist of the maximum number of water-table iterations, the maximum allowable displacement, and the closure tolerance for achieving steady state. The maximum number of water-table iterations is represented as program variable NITSW, and the maximum allowable displacement is represented as program variable DSMX. The closure tolerance for achieving steady state is represented as program variable TOLSW. Values for DSMX and TOLSW are selected so that an adequate steady-state solution is obtained within a minimum number of iterations. A discussion of values that are

normally input for these variables is given in the sections "Nonlinear Case" in Cooley (1992) and "Nonlinear Conditions" in Torak (1993).

The indicator variable, *ISTD*, described in the previous section, is used to indicate that the total known head change at specified head nodes is to be computed for nonlinear steady-state conditions. The indicator variable is input in subroutine DATIN, and computations for the total head change also are performed in this subroutine. Head changes at specified-head boundaries are represented as program vector DHB.

The displacements,  $\delta_l$ , in equation (5) are decreased, or damped, automatically during each water-table iteration in subroutine SWBDMP. The value of the maximum allowable displacement, DSMX, determines the amount of damping that is applied to the displacements. It is used in computing the damping parameter,  $\rho_l$ , which multiplies the displacements. The damping parameter is represented as program variable RP and is computed according to the three-step process given by equations (241)–(243) in Cooley (1992). The displacements are represented as program vector B. Other terms that are involved in computing RP are the maximum displacement and its absolute value for the current iteration level, *l*, and the ratio of damped-to-undamped displacements. These values are represented, respectively, as program variables DSP, DSPA, and SPR.

Hydraulic head for the advanced iteration level, *l*+1, is computed in subroutine HCALC according to equation (239) in Cooley (1992) by using the damped displacements. The updated heads are stored in program vector H and are used in subsequent computations for updating aquifer thickness or for determining the formulation of nonlinear boundary conditions (see previous sections for details about these computations).

## Changing Stresses and Boundary Conditions With Time

Inputs and computations for changing stresses and boundary conditions at the beginning of a time step or stress period are made in subroutines CBCHG, COCHG, GNCHG, and VNCHG. Changes are made by evaluating indicator variables that correspond to each stress or boundary condition that is allowed to vary in MODFE. If the value of an indicator variable equals the current time-step number, then changes are made to the corresponding stress or boundary condition. Subroutines and indicator variables that correspond to changing stresses and boundary conditions with time are listed in table 16.

Changes to most stresses and boundary conditions that are listed in table 16 take two time steps to

Table 16.—Program variables and subroutines used to change stresses and boundary conditions with time

Stress or boundary condition	Number of time steps to implement change	Indicator variable	Old value	New value	Subroutine
Source-bed head for transient leakage	1	NCBCH	--	HRJ	CBCHG
Point sources/sinks	2	NWCH	QOLD	QNEW	COCHG
Areally distributed recharge/discharge	2	NQCH	QOLD	QNEW	COCHG
Source-bed head for steady leakage	2	NHRCH	--	HR(J)	COCHG
Specified-flux boundary	2	NBQCH	--	QNEW	COCHG
Head-dependent (Cauchy-type) boundary	2	NBQCH	-- --	HK(J) HL(J)	COCHG
Specified-head boundary	1	NHCH	--	HB	COCHG
Nonlinear head-dependent (Cauchy-type) boundary	2	NGNCH	-- --	HRK(J) HRL(J)	GNCHG
Controlling head for nonlinear steady leakage	2	NVNCH	--	HS(J)	VNCHG

implement in MODFE because of the manner in which known terms in the B vector of matrix equations (1), (4), and (5) are represented. To allow for a time-varying B vector, components at the current and advanced time steps are needed to compute average values for the time step in which a change occurs (see development of a time-varying B vector,  $\bar{B}$ , in equations (60)–(62) in Cooley (1992)). However, to minimize computer storage, B-vector components at only one time step are represented in MODFE. Therefore, computations for components of the average B vector,  $\bar{B}$ , are made by the user, external to MODFE, according to equation (62) in Cooley (1992). Details of these computations are given in the section “Changing Stresses and Boundary Conditions With Time” in Torak (1993).

Changes to source-bed heads for transient leakage and to specified-head boundaries require one time step to implement. Computations for these features use the new values of head at the advanced time step to obtain the appropriate head changes for formulating matrix-equation components. For transient leakage, the head change is computed as program vector DHR and represents the difference in source-bed head for the two time steps; no average heads or head changes are used. For specified-head boundaries, the new values of head are used to compute average heads and

average head changes for the first time step that uses the new boundary head. The head change is represented by program vector DHB and the average head is stored in program vector H.

For changing values of point sources and sinks and of areally distributed recharge and discharge, the old (or current) values of stress are input along with the new values. The old values are required because known fluxes at a node are not stored individually in MODFE. Thus, the old values, input as program variable QOLD, are subtracted from the vector Q, which contains the known fluxes at a node. New values for changing stresses and boundary conditions with time are represented by the program variables listed in table 16.

## Water-Balance Summary and Flow Imbalance

A water-balance (mass-balance) summary and a flow imbalance are computed in MODFE by evaluating water-balance equations at each node that are based on equations (1) through (5). Volumetric flows [ $\text{length}^3/\text{time}$ ] which represent hydrologic processes in these equations are summed by node to obtain a quantitative summary of inflow and outflow rates for each time step (or for steady-state conditions) and of

total inflow and outflow volumes from the beginning of the simulation. A flow imbalance is computed by summing rates and volumes that correspond to hydrologic processes according to the system of nodal mass-balance equations given by equation (64) in Cooley (1992). The flow imbalance provides a quantitative measure of the numerical accuracy of the solution.

The matrix equations that are evaluated in MODFE for the water-balance summary are given in Cooley (1992) as equations (58), (83), (230), and (233). Components of these equations are defined in the development of equations (62)–(64) and in the discussion in the section “Extensions to the Basic Equations” in Cooley (1992). The matrix equations are restated here for reference. For linear, nonsteady-state conditions, the water-balance summary is based on equation (58) in Cooley (1992), which is restated as

$$\left( \frac{\underline{C}}{\begin{pmatrix} 2 \\ 3 \end{pmatrix} \Delta t_{n+1}} + \underline{A} \right) \underline{\delta} = \underline{B} + \underline{A} \hat{h}_n \quad (6)$$

For nonlinear, nonsteady-state conditions, the water-balance summary is given by equation (83) in Cooley (1992):

$$\frac{\underline{C}}{\begin{pmatrix} 2 \\ 3 \end{pmatrix} \Delta t_{n+1}} \underline{\delta} + (\underline{G} + \underline{V}) \bar{h} + (\underline{G} - \underline{G}) \underline{\delta} - \underline{B} \approx 0 \quad (7)$$

For linear, steady-state conditions, the matrix equation for the water-balance summary is given by equation (230) in Cooley (1992):

$$\underline{A} \hat{h} - \underline{B} \approx 0 \quad (8)$$

For nonlinear, steady-state conditions, equation (233) in Cooley (1992) is evaluated for the water-balance summary:

$$\underline{A}(\hat{h}) \hat{h} - \underline{B}(\hat{h}) \approx 0 \quad (9)$$

Each water-balance term contains program variables that represent inflow and outflow rates for the time step and total inflow and outflow volumes from the beginning of the simulation (table 17). Program variables for inflow have names that end with the letter “I”, and program variables for outflow have names that end with the letter “O”. An exception to this naming convention applies to water-balance terms for the accumulation of water in aquifer storage, where one program variable (SA) represents the

volumetric flow rate, and another (TSA) represents the total volume from the beginning of the simulation. Positive values of SA and TSA indicate an increase in water that is stored in the aquifer; negative values indicate the release of water from aquifer storage.

Program variables that represent total volumes of inflow or outflow from the beginning of the simulation for each water-balance term have names that begin with the letter “T” (table 17). These terms are computed by multiplying the volumetric rates, given by the corresponding program variables, by the time step size, and by summing the results for each time step.

Computations of water-balance terms that are contained in the C, G, and V matrices and B vector of equations (6) through (9) are performed by different subroutines according to the version of MODFE that is used. Program structures of each version and subroutine names are given in the section “Program Structures and Lists of Main Programs.” Matrix and vector components that are stored in program vectors A and Q are assembled by hydrologic term at each node for the water-balance computations. For linear conditions, formulation of components to the C, G, and V matrices and B vector follows identically from computations that were described in previous sections for forming coefficients and equations for hydrologic terms. For nonlinear conditions, formulation of these components differs from that used for linear conditions in that updated values of hydraulic head from either the previous iteration level (steady-state) or the corrector step (nonsteady-state) are used to evaluate and compute water-balance terms.

Hydraulic head is represented by program vector H for computations of the water-balance summary. For nonsteady-state conditions, program vector H represents the average hydraulic head,  $\bar{h}$  for the time step. For steady-state conditions, program vector H represents the steady-state solution,  $\hat{h}$ . Both  $\bar{h}$  and  $\hat{h}$  are computed in subroutine HCALC prior to computing the water-balance summary.

The average head change,  $\delta$ , of equation (57) in Cooley (1992) and appearing in equations (6) and (7), is represented by different program variables for computation of the water-balance summary depending on the version of MODFE that is used. For linear and nonlinear nonsteady-state versions, program vector DHB represents average head changes at specified-head nodes. For linear nonsteady-state conditions, program vector B represents average head changes at all other nodes. For nonlinear, nonsteady-state conditions, program vector DH represents average head changes at all nodes except specified-head nodes. The head changes are used to compute water-balance terms for the accumulation of water in aquifer

Table 17.—Program variables and subroutines used to compute water-balance summary and flow imbalance

Water-balance term	Subroutine(s)	Program variable(s)	Component; matrix equation
Accumulation of water in aquifer storage	MASBAL, MBALCB MBALWT, MBWTCB MASOUT	SA TSA	C matrix; (6) and (7)
Steady vertical leakage	MASBAL, MBALCB MBALWT, MBWTCB MASOUT	VLQI, VLQO TLQI, TLQO	V matrix and B vector; (6)-(9)
Transient leakage	MBALCB, MBWTCB MASOUT	VLQI, VLQO TLQI, TLQO	V matrix and B vector; (6) and (7)
Areally distributed sources and sinks	FMCO MASOUT	DQI, DQO TDQI, TDQO	B vector; (6)-(9)
Point sources and sinks	DATIN MASOUT	WQI, WQO TWQI, TWQO	B vector; (6)-(9)
Specified-head boundaries	MASOUT	HBQI, HBQO THBQI, THBQO	All components; (6)-(9)
Specified-flux boundaries	MASBAL, MBALCB MBALWT, MBWTCB MASOUT	BQI, BQO TBQI, TBQO	B vector; (6)-(9)
Head-dependent (Cauchy-type) flux boundaries	MASBAL, MBALCB MBALWT, MBWTCB MASOUT	BQI, BQO TBQI, TBQO	V matrix and B vector; (6)-(9)
Nonlinear, head-dependent (Cauchy-type) flux	GNBAL, GNBLS	BNQI, BNQO TBNQI, TBNQO	V matrix and B vector; (7)-(9)
Nonlinear point sinks	GNBAL, GNBLS	PNQO, TPNQO	V matrix and B vector; (7)-(9)
Nonlinear, steady vertical leakage	VNBAL, VNBLS	VNLQI, VNLQO TNLQI, TNLQO	V matrix and B vector; (7)-(9)
Flow imbalance	MASBAL, MBALCB MBALWT, MBWTCB GNBAL, GNBLS VNBAL, VNBLS MASOUT	ER TER	G and V matrices and B vector, (7)-(9)

storage, to update aquifer thickness for transmissivity computations, and to compute aquifer head at the end of the time step for evaluating nonlinear water-balance terms.

The head change from the initial condition is represented by program vector B for the water-balance of nonlinear steady-state conditions, equation (9). These head changes are used to update aquifer thickness for transmissivity computations and to compute the head at the end of the water-table iteration for evaluating nonlinear water-balance terms.

Flow across specified-head boundaries is computed for the water-balance summary by evaluating equations (6) through (9) for the patch of elements that is centered on each specified-head node. The computed head at nodes within the patch and the specified head are used to compute a flow balance at each specified-

head node. The computations are performed by the subroutines indicated in table 17, with the net flow for each specified-head boundary represented by program vector R. Values in the R vector are summed according to sign (positive for inflow or recharge) to obtain the terms, HBQI and HBQO, in the water-balance summary for volumetric rates of recharge and discharge, respectively. The flow rates are multiplied by the time-step size and summed over all time steps to obtain the total volume of water that recharges (THBQI) or discharges (THBQO) across specified-head boundaries from the beginning of the simulation. For steady-state simulations, the time-step size is set to unity (=1), and only one time step is used; thus, the flow rates, HBQI and HBQO, and the volumes, THBQI and THBQO represent inflows and outflows, respectively, for one time unit.

The water-balance summary can be used to determine the accuracy at which matrix equations (1) through (5) had been solved during the simulation. A printout of the water-balance summary is made after each time step or at the end of the water-table iteration. For linear conditions, the flow imbalance within the water-balance summary indicates the accuracy of the solution method used by MODFE to solve the matrix equations. Acceptable values for the flow imbalance, program variable ER, are about 5 or 6 orders of magnitude smaller than the largest value among the other water-balance terms. Values for ER within this range indicate that the matrix equations were solved to within the limit of accuracy permitted by the computer.

The flow imbalance within the water-balance summary gives a quantitative measure of the accuracy in approximating nonlinear hydrologic processes using MODFE. For nonlinear conditions, differences can exist between values in the C, G, and V matrices and B vector that are computed for the last water-table iteration or corrector step and values that are computed for the water-balance summary. These differences are manifested in the flow imbalance, where values for ER or TER that are larger than the criterion given above can exist if either the predictor-corrector technique or the water-table iteration cannot approximate accurately the nonlinear processes within the aquifer problem. Usually, large flow imbalances can be decreased to acceptable values by performing the same simulation with smaller time-step sizes, which are contained in program vector DT (nonsteady state), or a decreased error tolerance, program variable TOLSW (steady state). Details of inputting these values are given in the section "Input Instructions" in Torak (1993).

## Computational Aspects of Solution Methods

### Direct—Symmetric-Doolittle Decomposition

Direct solution of finite-element matrix equations (1)–(5) by symmetric-Doolittle decomposition (see section "Symmetric-Doolittle Method," in Cooley, 1992) is implemented in MODFE by subroutines INITB, SETB, and BAND. Inputs are made to subroutine INITB that are used to determine the number of equations that are solved and to allocate computer storage for solution. These inputs define the number of nodes, number of specified-head boundaries, maximum condensed-matrix bandwidth, and the maximum

reduced-matrix bandwidth, and are represented by program variables NNDS, NHDS, MBWC, and MBW, respectively. Descriptions of the maximum condensed-matrix bandwidth and the maximum reduced-matrix bandwidth are given in the section "Node Numbering and Determining Bandwidth," in Torak (1993). A discussion of the effects of the bandwidth determinations on computer storage is given in the section "Allocation of Computer Storage and Processing Time."

Computations for determining the maximum reduced-matrix bandwidth and the writing of model information to computer files are performed in subroutine SETB. Although a value for the maximum reduced-matrix bandwidth is input as program variable MBW, this value is used only as an estimate for allocating computer storage. The actual value for the maximum reduced-matrix bandwidth is determined in subroutine SETB and is represented by program variable IBND. The value for IBND is used to compute the amount of storage locations that is overwritten during solution by the direct method. Components of the C, V, and G matrices and B vector of equations (1)–(5), element areas, and node incidences are written to computer files by this subroutine to enable overwriting of storage locations during solution. The process of overwriting storage locations by the solution methods is described in the section "Solution Methods." Other functions of subroutine SETB are outlined in table 15 and are indicated by comment statements within the subroutine (see section "List of Subroutines").

Matrix equations (1)–(5) are solved in subroutine BAND. The solution procedure is given by equations (263)–(265) in Cooley (1992), and consists of three steps: factorization, equation (263); forward substitution, equation (264); and back substitution, equation (265). These steps are outlined by comment statements within the subroutine (see section "List of Subroutines").

### Iterative—Modified Incomplete-Cholesky Conjugate Gradient

The iterative, modified incomplete-Cholesky conjugate gradient (MICCG) method of solution is implemented in MODFE by subroutines INITCG, SETCG, and MICCG. Inputs associated with the MICCG method are made in subroutine INITCG. These inputs consist of values for the maximum number of iterations and the closure tolerance, and are represented by program variables NIT and TOL, respectively. Variables that define the beginning of storage locations within the general storage vector G also are

evaluated in subroutine INITCG. Details about the allocation of computer storage in MODFE are given in the following section.

Components of the C, V, and G matrices and B vector of equations (1)–(5), element areas, and node incidences are written to computer files by subroutine SETCG. This process enables storage locations that are occupied by these terms to be overwritten during solution by the MICCG method. Overwriting storage locations by the solution methods is described in the section “Solution Methods.”

Computations for solving equations (1)–(5) by the MICCG method are contained in subroutine MICCG. The solution process begins by approximately factoring the reduced matrix into an upper triangular matrix, U, which is represented by program vector AF. The factorization is performed according to equation (274) in Cooley (1992), and is modified by the row-sums agreement given by equations (276)–(278) in Cooley (1992). An intermediate vector,  $\bar{y}$ , is computed by forward substitution, and the displacement vector, s, is computed by backward substitution according to equations (283) and (284) in Cooley (1992). These computations are identified in the subroutine by comment statements.

Computations for the generalized conjugate-gradient algorithm of equation (271) in Cooley (1992) are performed in subroutine MICCG by formulating iteration parameters  $\beta_k$  and the A-orthogonal vectors,  $\underline{p}_k$ . The iteration parameter used for each vector is represented by program variable S, and the vectors are represented by program variable P. The scaled displacements,  $\alpha_k \underline{p}_k$ , are represented by program variable TMPA, and are summed for each equation and stored in program variable B to give the total displacement from the initial condition. Similarly, scaled residuals,  $-\alpha_k \underline{A} \underline{p}_k$ , are computed for each equation, and the sum is stored in program variable R. The largest absolute value of the displacements is determined according to equation (285) in Cooley (1992) by comparing absolute values of TMPA, and the result is represented as program variable PMAX. Similarly, the largest absolute scaled residual, given by equation (289) in Cooley (1992) is determined, and the result is represented by program variable RMAX. The iterative process is concluded (has converged) when values of RMAX and PMAX are less than the closure tolerance, which is represented by program variable TOL.

Messages are printed out from subroutine MICCG that describe the progress toward attaining a solution. If convergence is achieved, then a message is printed to that effect along with the number of iterations that was required. If convergence was not achieved, then a message is printed indicating that the solution failed

to converge, followed by values for the maximum number of iterations (NIT), the largest absolute scaled residual (RMAX), and the computed head changes (program vector B).

## Allocation of Computer Storage and Processing Time

Several techniques for storing and representing program variables have been used during the design of MODFE to minimize computer storage and processing time. A single program vector, G, which is dimensioned at the time of execution of MODFE, is used to store most of the information that is needed to conduct a simulation. Components of matrix equations (1)–(5) are stored within vector G in a manner that decreases the range of indexes that are searched by the computer during assembly of the matrix equations. The order of the coefficient matrix is decreased by eliminating specified-head nodes from the solution process, thereby allowing a reduced-matrix equation to be solved. Use of the reduced matrix for solution results in fewer equations and storage locations than if specified-head nodes were retained in the formulation. Components of the reduced matrix are stored in a condensed form, thus eliminating as many zero entries from storage as possible. In addition, storage locations within program vector G are overwritten, or reused, routinely during solution of the matrix equations. Details of these techniques are given in the following sections.

### General-Storage Vector G

Nearly all program information that is needed to form and solve matrix equations (1)–(5) and to compute a water-balance summary is contained in the general-storage vector G. The vector G is subdivided into smaller vector lengths of storage, and all values are stored in single-subscripted form. Computer storage is allocated at the time MODFE is executed by determining the beginning locations (starting addresses) of the vector lengths that correspond to program vectors stored in general-storage vector G. Values for program variables that represent starting addresses are computed automatically by MODFE from inputs that define program dimensions and problem specifications (see section “Input Instructions” in Torak (1993)). Descriptions of the terms for which computer storage is allocated within the general-storage vector G and names of subroutines that perform the storage allocation are given in table 6. Program variables that represent starting addresses,